

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-249910

(43)Date of publication of application : 14.09.2001

(51)Int.Cl.

G06F 15/177  
 G06F 9/445  
 G06F 9/46  
 G06F 13/00  
 G06F 15/16  
 G06F 17/60

(21)Application number : 2000-058372

(71)Applicant : KDDI CORP

(22)Date of filing : 03.03.2000

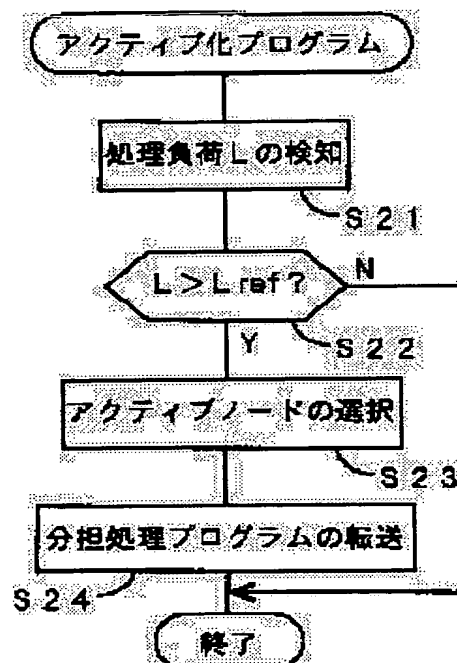
(72)Inventor : YOSHIHARA TAKAHITO  
 HATTORI HAJIME  
 SUGIYAMA KEIZO  
 NAKAO KOJI

## (54) METHOD FOR ARRANGING PROGRAM TO NODE ON NETWORK

(57)Abstract:

PROBLEM TO BE SOLVED: To dynamically optimize the number and position of active nodes corresponding to the processing loads of a processing terminal and the active node.

SOLUTION: This method for arranging a program to a node on a network for feeding a partial charge processing program to the node of the network provided with the node capable of executing the fed program, turning it to the active node and making the active node take the partial charge of a part of a processing to be executed by a terminal is provided with a procedure (steps S21 and S22) for discriminating whether or not the load of the processing in at least one of the terminal and the active node reaches a scheduled reference load and the procedure (steps S23 and S24) for feeding the partial charge processing program to at least one node on the network and turning it to the active node when the processing load reaches the scheduled reference load.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision  
of rejection]

[Date of requesting appeal against examiner's  
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号  
特開2001-249910  
(P2001-249910A)

(43) 公開日 平成13年9月14日 (2001.9.14)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テーマコード* (参考)
G 0 6 F 15/177	6 7 4	G 0 6 F 15/177	6 7 4 B 5 B 0 4 5
	6 7 0		6 7 0 B 5 B 0 4 9
9/445		9/46	3 6 0 C 5 B 0 7 6
9/46	3 6 0	13/00	3 5 7 Z 5 B 0 8 9
13/00	3 5 7	15/16	6 2 0 B 5 B 0 9 8

審査請求 未請求 請求項の数 8 O L (全 7 頁) 最終頁に続く

(21) 出願番号 特願2000-58372(P2000-58372)

(22) 出願日 平成12年3月3日 (2000.3.3)

(71) 出願人 000208891

ケイディーディーアイ株式会社  
東京都新宿区西新宿二丁目3番2号

(72) 発明者 吉原 貴仁

埼玉県上福岡市大原2-1-15 株式会社  
ケイディディ研究所内

(72) 発明者 服部 元

埼玉県上福岡市大原2-1-15 株式会社  
ケイディディ研究所内

(74) 代理人 100084870

弁理士 田中 香樹 (外1名)

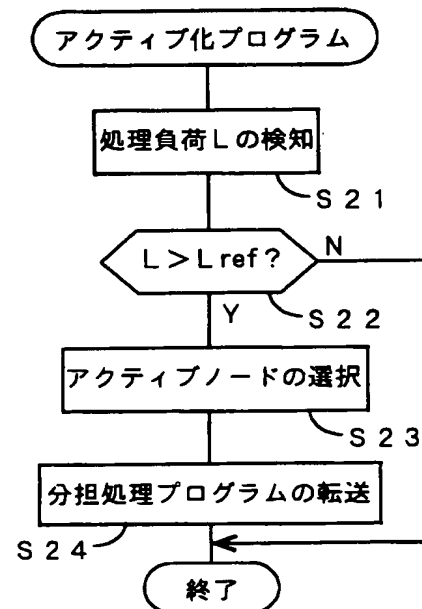
最終頁に続く

(54) 【発明の名称】 ネットワーク上のノードへのプログラム配置方法

(57) 【要約】

【課題】 処理端末やアクティブノードの処理負荷に応じて、アクティブノードの個数や位置を動的に最適化する。

【解決手段】 投入されたプログラムを実行可能なノードを含むネットワークの当該ノードに分担処理プログラムを投入してこれをアクティブノード化し、端末が実行すべき処理の一部を前記アクティブノードに分担させるネットワーク上のノードへのプログラム配置方法において、前記端末およびアクティブノードの少なくとも一方における前記処理の負荷が予定の基準負荷に達しているか否かを判別する手順 (ステップ S 2 1, 2 2) と、前記処理負荷が予定の基準負荷に達すると、ネットワーク上の少なくとも一つのノードに、前記分担処理プログラムを投入してアクティブノード化する手順 (ステップ S 2 3, 2 4) とを含む。



**【特許請求の範囲】**

**【請求項1】** 投入されたプログラムを実行可能なノードを含むネットワークの当該ノードに分担処理プログラムを投入してこれをアクティブノード化し、端末が実行すべき処理の一部を前記アクティブノードに分担させるネットワーク上のノードへのプログラム配置方法において、

前記端末およびアクティブノードの少なくとも一方における前記処理の負荷が予定の基準負荷に達しているか否かを判別する手順と、

前記処理負荷が予定の基準負荷に達すると、ネットワーク上の少なくとも一つのノードに、前記分担処理プログラムを投入してアクティブノード化する手順とを含むことを特徴とするネットワーク上のノードへのプログラム配置方法。

**【請求項2】** 前記端末は、自身の処理負荷が予定の基準負荷に達すると、ネットワーク上の少なくとも一つのノードに前記分担処理プログラムを投入し、これをアクティブノード化することを特徴とする請求項1に記載のネットワーク上のノードへのプログラム配置方法。

**【請求項3】** 前記アクティブノードは、自身の処理負荷が予定の基準負荷に達すると、ネットワーク上の少なくとも一つのノードに前記分担処理プログラムを投入し、これをアクティブノード化することを特徴とする請求項1または2に記載のネットワーク上のノードへのプログラム配置方法。

**【請求項4】** 前記ノードは、自身が過負荷状態であると、前記分担処理プログラムを自身に配置させないことを特徴とする請求項1ないし3のいずれかに記載のネットワーク上のノードへのプログラム配置方法。

**【請求項5】** 前記ノードは、自身に投入された分担処理プログラムを配置することなく他のノードに再投入し、これをアクティブノード化することを特徴とする請求項4に記載のネットワーク上のノードへのプログラム配置方法。

**【請求項6】** 前記各アクティブノードに配置された分担処理プログラムは、当該分担処理プログラムの利用頻度が小さい場合に削除されることを特徴とする請求項1ないし5のいずれかに記載のネットワーク上のノードへのプログラム配置方法。

**【請求項7】** 前記端末およびアクティブノードは、自身と隣接する他のノードに分担処理プログラムを追加的に投入することを特徴とする請求項1ないし6のいずれかに記載のネットワーク上のノードへのプログラム配置方法。

**【請求項8】** 前記端末およびアクティブノードは、自身と隣接する他のノードに分担処理プログラムを投入し、自身の分担処理プログラムは削除することを特徴とする請求項1ないし6のいずれかに記載のネットワーク上のノードへのプログラム配置方法。

**【発明の詳細な説明】****【0001】**

**【発明の属する技術分野】** 本発明は、端末同士がノードを介して接続されるネットワークに係り、特に、端末が実行すべき処理の一部をノードに分担させることで端末の処理負荷を軽減するアクティブネットワーク技術に好適な、ネットワーク上のノードへのプログラム配置方法に関する。

**【0002】**

10 **【従来の技術】** 端末同士が複数のノードを介して接続されたネットワークにおいて、端末が実行すべき処理の一部を各ノードに分担させるための分担処理プログラムを、当該ノードに投入かつ配置（以下、アクティブ化と表現する場合もある）し、アクティブ化されたノード（以下、アクティブノードと表現する場合もある）に分担処理プログラムを実行させるアクティブネットワーク技術が提案されている。このようなアクティブネットワーク技術によれば、端末の処理負荷が各アクティブノードに分散されるので、当該端末の処理負荷を軽減すること  
20 ができる。

**【0003】**

**【発明が解決しようとする課題】** 上記したアクティブネットワークでは、分担処理プログラムを格納したアクティブノードを多数配置するほど端末の負担が軽減される反面、多数のアクティブノードがパケットやパケット内のデータに対して分担処理プログラムを実行することになる。このため、送受信端末間での伝送遅延は、端末の負荷が軽減されることによる短縮分と、アクティブネットワーク内で分担処理プログラムの実行回数が増加することによる増加分との差分だけ改善されることとなる。したがって、分担処理プログラムを投入してアクティブ化するノードの数は、端末やアクティブノードの処理負荷が過大とならない範囲で最小に抑えることが望ましい。

30 **【0004】** さらに、アクティブノードはパケットの入力が集中する位置に配置することが望ましく、パケットが集中しないノードに分担処理プログラムを配置してアクティブ化しても、十分な改善効果を得ることはできない。

**【0005】** しかしながら、端末やアクティブノードの処理負荷、あるいはパケットの集中箇所等は逐次変化するため、最適なアクティブノード数や配置位置を予め設定することは難しい。このため、従来のアクティブネットワーク技術では、アクティブ化するノードの位置や個数の最適解を、シミュレーション結果や経験則に基づいて予測し、この予測結果に基づいて所定のノードを固定的にアクティブノード化していた。すなわち、端末や各ノードの処理負荷が変化しても、最初に配置したアクティブノードの位置や個数がそのまま維持され続けていた。

50 **【0006】** このように、従来のアクティブネットワー

ク技術では、端末やアクティブノードでの実際の処理負荷とは無関係に、所定のノードが固定的にアクティブノード化されていたため、ネットワーク全体の処理負荷や伝送遅延を効率良く改善することが難しかった。

【0007】本発明の目的は、上記した従来技術の課題を解決し、時々刻々と変化する処理端末やアクティブノードの処理負荷に応じて、アクティブ化するノードの個数や位置を動的に最適化できるネットワーク上のノードへのプログラム配置方法を提供することにある。

【0008】

【課題を解決するための手段】上記した目的を達成するために、本発明は、投入されたプログラムを実行可能なノードを含むネットワークの当該ノードに分担処理プログラムを投入してこれをアクティブノード化し、端末が実行すべき処理の一部を前記アクティブノードに分担させるネットワーク上のノードへのプログラム配置方法において、前記端末およびアクティブノードの少なくとも一方における前記処理の負荷が予定の基準負荷に達しているか否かを判別する手順と、前記処理負荷が予定の基準負荷に達すると、ネットワーク上の少なくとも一つのノードに、前記分担処理プログラムを投入してアクティブノード化する手順とを含むことを特徴とする。

【0009】上記した特徴によれば、端末やアクティブノードの処理負荷が高くなったときに初めて、他のノードに分担処理プログラムが配置されて新たにアクティブノード化され、当該新たなアクティブノードがデータ処理の一部をさらに分担する。したがって、アクティブ化するノードの数を、端末やアクティブノードの処理負荷が過大とならない範囲で最小に抑えることができる。

【0010】

【発明の実施の形態】以下、図面を参照して本発明を詳細に説明する。図1は、本発明を適用したアクティブネットワークの構成を示した図であり、ここでは、ネットワークを例にして本発明の実施形態を説明する。

【0011】受信端末としてのオークション端末10は、少なくとも1つのノードN(N1, N2, …Nn)を含むネットワークNTを介して、送信端末としての複数のクライアント端末S(S1, S2 …Sn)と接続されている。各クライアント端末Sから入力された入札希望価格Yinは、前記ネットワークNTを介してオークション端末10へ入力される。前記オークション端末10は、各クライアント端末Sから入力される入札希望価格Yinを比較し、最高価格を入札したクライアント端末に対して購入権を与える。

【0012】図2は、上記したオークション端末10の主要部の構成を示した機能ブロック図であり、当該オークション端末10をネットワークNTに接続するための通信インターフェース101と、ROM102に格納されたプログラムにしたがって所定の処理を実行するCPU103と、前記CPU103のワークエリアとして機

能するRAM104とを少なくとも含む。

【0013】前記ROM102には、当該オークション端末10が通信端末として機能するための基本プログラムの他に、上記したネットワークNTを実行するためのオークションプログラム102aと、当該オークション端末10の処理負荷を前記CPU103の稼働率やRAM104の占有率、あるいは通信インターフェース101の送受信バッファのキュー長等に基づいて判定し、処理負荷が高い場合にネットワークNT内のノードNを動的にアクティブノード化するアクティブ化プログラム102bとが格納されている。

【0014】前記RAM104には、上記したワークエリアとは別に、ネットワークNT内のノードNをアクティブ化するために、当該ノードNに対して転送される分担処理プログラム104aが格納されている。

【0015】次いで、図3のフローチャートを参照して、上記したオークションプログラム102aの動作について説明する。

【0016】ステップS11において、いずれかのクライアント端末Sから送出された入札希望価格Yinを検知すると、ステップS12では、これまでに入力された入札希望価格Yinの最高額である最高価格Ymaxと、前記検知された今回の入札希望価格Yinとが比較される。入札希望価格Yinが最高価格Ymax以下であれば、ステップS13において、当該入札希望価格Yinを送出したクライアント端末Sに対して、当該入札希望価格Yinが最高価格Ymax以下である旨を表す失敗メッセージが送出される。

【0017】また、前記ステップS12において、入札希望価格Yinが最高価格Ymaxを上回っていれば、ステップS14において、現在の最高価格Ymaxが今回の入札希望価格Yinに更新される。ステップS15では、更新された最高価格Ymaxが、ネットワークNT上の各ノードNを経由して全クライアント端末Sへ通知される。これにより、各クライアント端末Sでは現在の最高価格Ymaxを認識できるようになる。

【0018】次いで、図4のフローチャートを参照して、上記したアクティブ化プログラム102bの動作について説明する。

【0019】ステップS21では、CPU103の稼働率やRAM104の占有率、あるいは通信I/F101の送受信バッファのキュー長等に基づいて、当該オークション端末10の現在の処理負荷Lが検知される。ステップS22では、検知された現在の処理負荷Lと基準負荷Lrefとが比較される。

【0020】ここで、図5に示したように、オークションに参加しているクライアント端末Sが少なく、例えば、クライアント端末S2, S8のみであれば、オークション端末10の処理負荷は低いままである。したがって、ステップS22の判断は否定となり、いずれのノード

ドNもアクティブ化されることはない。このため、オークション端末10は各クライアント端末S4, S8から転送された入札希望価格Y<sub>in</sub>を、上記したオークションプログラム102aにしたがって全て自身で順次処理することになる。

【0021】これに対して、図6に示したように、オークションに参加するクライアント端末Sの数が増大し、前記ステップS22において、処理負荷Lが基準負荷L<sub>ref</sub>を超えたと判断されると、ステップS23において、アクティブ化すべきノードNが選択される。本実施形態では、以下の理由から、オークション端末10に隣接するノードN1が選択される。

【0022】すなわち、アクティブノードをオークション端末10に隣接配置した場合と離間配置した場合とで、オークション端末10から送出される最高価格Y<sub>max</sub>の更新結果の到達時刻を比較すると、隣接配置されたアクティブノードへは、離間配置されたアクティブノードよりも早く到達する。したがって、隣接ノードをアクティブ化すれば、離間ノードをアクティブ化する場合に比べて、より多くの入札希望価格をフィルタリングすることができ、オークション端末10の負荷を、より多く軽減できるからである。

【0023】ステップS24では、前記選択されたノードN1に対して、前記RAM104に格納されている分担処理プログラム104aが投入される。

【0024】分担処理プログラム104aを投入されたノードN1はアクティブ化(AN1)され、以下に図7のフローチャートを参照して詳述するように、当該分担処理プログラム104aを実行してオークション端末10の処理を分担するので、オークション端末10の処理負荷が低減される。

【0025】図7は、前記分担処理プログラム104aを投入されたアクティブノードAN1が、当該分担処理プログラム104aに従って実行する分担処理の動作を示したフローチャートである。

【0026】分担処理プログラム104aが起動されると、本実施形態では、ステップS30～S38の『オークション処理』と、ステップS40～S43の『負荷判定処理』とが並列的に実行される。

【0027】『オークション処理』では、ステップS30でアンロードタイマT<sub>m</sub>がスタートする。このアンロードタイマT<sub>m</sub>は、当該分担処理プログラム104aが十分に活用されないにもかかわらずノードNに常駐することを防止するために設けられ、処理対象の入札希望価格Y<sub>in</sub>が入力されない経過時間を計測する。

【0028】ステップS31において、いずれかのクライアント端末Sから送出された入札希望価格Y<sub>in</sub>を検知すると、ステップS32では、前記アンロードタイマT<sub>m</sub>がリセットされる。ステップS33では、当該アクティブノードAN1における現在までの最高価格Y<sub>max1</sub>と

前記検知された入札希望価格Y<sub>in</sub>とが比較され、入札希望価格Y<sub>in</sub>が最高価格Y<sub>max1</sub>以下であれば、ステップS38において、当該入札希望価格Y<sub>in</sub>を送出したクライアント端末Sに対して失敗メッセージが送出される。

【0029】また、前記ステップS33において、入札希望価格Y<sub>in</sub>が最高価格Y<sub>max1</sub>を上回ったと判定されると、ステップS34では、現在の最高価格Y<sub>max1</sub>が今回の入札希望価格Y<sub>in</sub>に更新される。ステップS35では、更新された現在の最高価格Y<sub>max1</sub>がオークション端末10へ転送される。

【0030】一方、前記ステップS31において、入札希望価格Y<sub>in</sub>が検知されないと、ステップS36では、前記アンロードタイマT<sub>m</sub>と基準時間T<sub>ref</sub>とが比較され、アンロードタイマT<sub>m</sub>が基準時間T<sub>ref</sub>を下回っている限りは、前記ステップS31へ戻って上記した各処理が繰り返される。また、アンロードタイマT<sub>m</sub>が基準時間T<sub>ref</sub>を上回ると、ステップS37において、当該分担処理プログラムが削除され、アクティブノードAN1は通常のノードN1に戻る。

【0031】一方、『負荷判定処理』では、ステップS40において、当該アクティブノードAN1の現在の処理負荷Lが、CPUの稼働率やメモリの占有率、あるいは送受信バッファのキュー長等に基づいて検知される。ステップS41では、検知された現在の処理負荷Lと基準負荷L<sub>ref</sub>とが比較される。

【0032】ここで、図8に示したように、オークション参加者が増加してアクティブノードAN1の処理負荷も増大し、現在の処理負荷Lが基準負荷L<sub>ref</sub>を超えたと判定されると、ステップS42において、アクティブ化すべきノードとして、隣接するノードN2が選択される。ステップS43では、ノードN2に対して分担処理プログラムが投入されてアクティブノード化される。

【0033】この結果、アクティブノードAN2からAN1へは、現在の最高価格以下の入札希望価格が入力されなくなるので、当該アクティブノードAN1の処理負荷が大幅に低減されることになる。

【0034】上記したように、本実施形態によれば、受信端末としてのオークション端末10や、既にアクティブ化されたノード(アクティブノードAN)の処理負荷を監視し、処理負荷が大きくなった場合のみ、所定のノードNをアクティブ化して処理を分散するので、必要以上に多くのノードがアクティブ化されることにより生じ得る伝送遅延の増大を防止しながら、受信端末やノードの負荷が軽減されることによる伝送遅延の解消が可能となり、総合的に伝送遅延を大幅に改善することができる。

【0035】また、本実施形態では、受信端末あるいはアクティブノードの処理負荷が増加してノードNのいずれかをアクティブ化する際に、隣接したノードから順にアクティブ化するようにしたので、アクティブノードにおいて生じ得る伝送遅延を最小限に抑えながら、ノード

10

20

30

40

50

のアクティブ化による負荷の分散効果により、過大な処理負荷が原因で生じていた伝送遅延を解消できるようになる。

【0036】さらに、本実施形態では、アクティブ化されたノードがアクティブノードとしての機能を十分に発揮していない場合には、分担処理プログラムをアンロードして通常のノードへ戻すようにしたので、処理負荷の分散に寄与しないアクティブノードが分担処理プログラムを無駄に実行することによる伝送遅延の発生を防止できるようになる。

【0037】なお、本発明は上記した実施形態に限定されるものではなく、以下に説明するように、種々のオプションを設定することも可能である。

【0038】上記した実施形態では、処理負荷の高くなったオークション端末あるいはアクティブノードANが他のノードに分担処理プログラムを投入してアクティブノードANを増やすものとして説明した。しかしながら、例えば図6に示したように、ノードN1がアクティブ化されているときに当該アクティブノードAN1の処理負荷が過大となった際、ノードN2を追加的にアクティブ化するのではなく、図9に示したように、分担処理プログラムを他のノードN2に移動、すなわち分担処理プログラムをノードN2へ投入すると同時に自身の分担処理プログラムを消去し、自らは通常のノードに戻るようにしても良い。

【0039】さらに、上記した実施形態では、オークション端末10やアクティブノードANから分担処理プログラムを投入されたノードが、無条件でアクティブ化されるものとして説明した。しかしながら、分担処理プログラムを投入されたノードNが自身の処理負荷Lを検知し、処理負荷Lが既に高い場合等には分担処理プログラムの配置を拒否したり、あるいは投入された分担処理プログラムを、処理負荷の低い他のノードへ再投入するようにしても良い。

【0040】さらに、上記した実施形態では、オークション端末やアクティブノードANが、隣接するノードNへ分担処理プログラムを投入してアクティブ化するものとして説明した。しかしながら、本発明はこれのみに限定されるものではなく、隣接以外の任意のノードNをアクティブ化しても良い。

【0041】例えば、図10に示したように、クライアント端末に近いノードN6に入力が集中しているような場合には、隣接するノードN1をアクティブ化するよりも、当該ノードN6をアクティブ化の方が効率的である。したがって、各パケットが経由したノードの履歴を当該パケットのヘッダ等に全て登録し、分担処理プログラムを投入しようとするオークション端末10あるいはアクティブノードANが受信パケットのヘッダを参照し、最も多く登録されているノードN3に分担処理プログラムを投入してアクティブ化するようにしても良い。

【0042】さらに、上記した実施形態では、本発明の理解を容易なものとするためにネットオークションを例にして説明した。しかしながら、本発明はこれのみに限定されるものではなく、インターネット等で広く普及しているクライアント/サーバ型のアプリケーション、たとえばWWW(World Wide Web)サーバやキャッシュサーバの処理負荷を軽減する目的での適用も可能であり、その適用範囲は極めて広い。

【0043】

【発明の効果】本発明によれば、以下のような効果が達成される。

【0044】(1) 端末あるいはアクティブノードの処理負荷を監視し、処理負荷が過大となった場合のみ、ネットワーク上の少なくとも一つの他のノードをアクティブノード化するので、アクティブノードの数を、端末やアクティブノードの処理負荷が過大とならない範囲で最小に抑えることができる。したがって、アクティブノードが分担処理プログラムを実行することにより生じ得る伝送遅延を最小限に抑えながら、ノードのアクティブ化による負荷の分散効果により、過大な処理負荷が原因で生じていた伝送遅延を解消できる。

【0045】(2) 端末あるいはアクティブノードの処理負荷が過大になると、隣接(ないしは近接)するノードをアクティブ化するようにしたので、隣接しない(ないしは近接しない)ノードをアクティブ化する場合に比べて、新たなアクティブノードの分担比率を増加させることができ、端末あるいは既存のアクティブノードの処理負荷を大幅に減じることが可能になる。

【0046】(3) アクティブ化されたノードがアクティブノードとしての機能を十分に発揮していない場合には、分担処理プログラムを削除して通常のノードへ戻すようにしたので、処理負荷の分散に寄与しないアクティブノードが分担処理プログラムを無駄に実行することによる伝送遅延の発生を防止できるようになる。

【図面の簡単な説明】

【図1】本発明を適用したアクティブネットワークの構成を示した図である。

【図2】本発明を適用した受信端末の機能ブロック図である。

【図3】オークションプログラムの動作を示したフローチャートである。

【図4】アクティブ化プログラムの動作を示したフローチャートである。

【図5】本発明によるアクティブネットワークの構築過程(その1)を模式的に示した図である。

【図6】本発明によるアクティブネットワークの構築過程(その2)を模式的に示した図である。

【図7】分担処理プログラムの動作を示したフローチャートである。

【図8】本発明によるアクティブネットワークの構築過

程（その3）を模式的に示した図である。

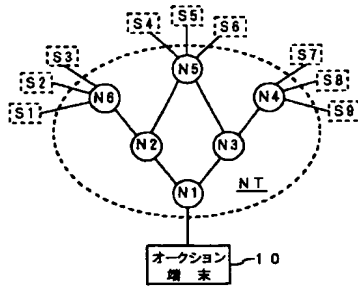
【図9】本発明によるアクティブネットワークの構築過程（その4）を模式的に示した図である。

【図10】本発明によるアクティブネットワークの構築過程（その5）を模式的に示した図である。

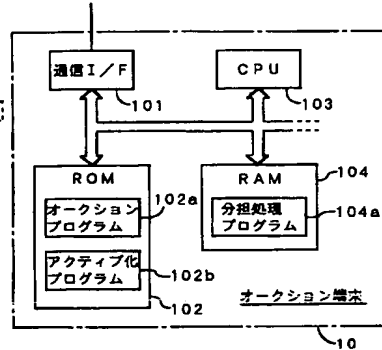
【符号の説明】

10…オークション端末、101…通信インターフェース、102…ROM、102a…オークションプログラム、102b…アクティブ化プログラム、103…CPU、104…RAM、104a…分担処理プログラム、N…ノード、NT…ネットワーク、端末S…クライアント

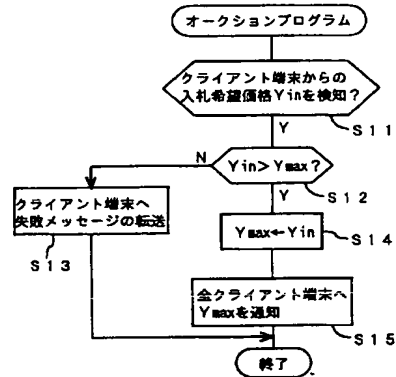
【図1】



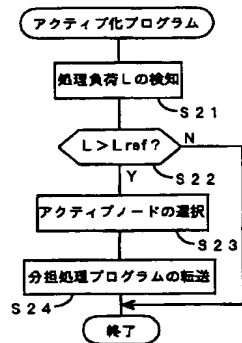
【図2】



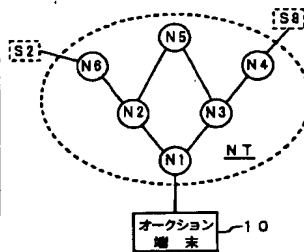
【図3】



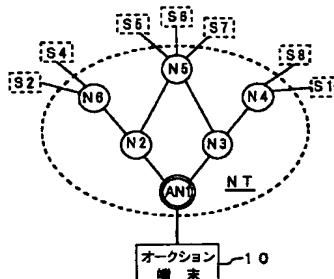
【図4】



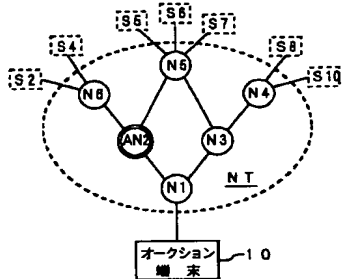
【図5】



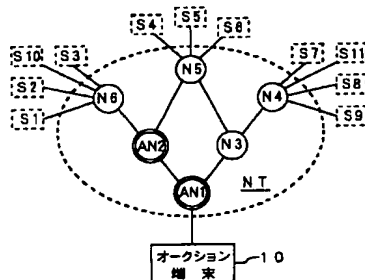
【図6】



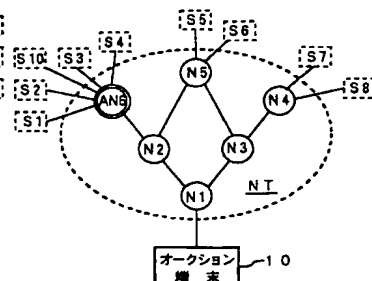
【図9】



【図8】

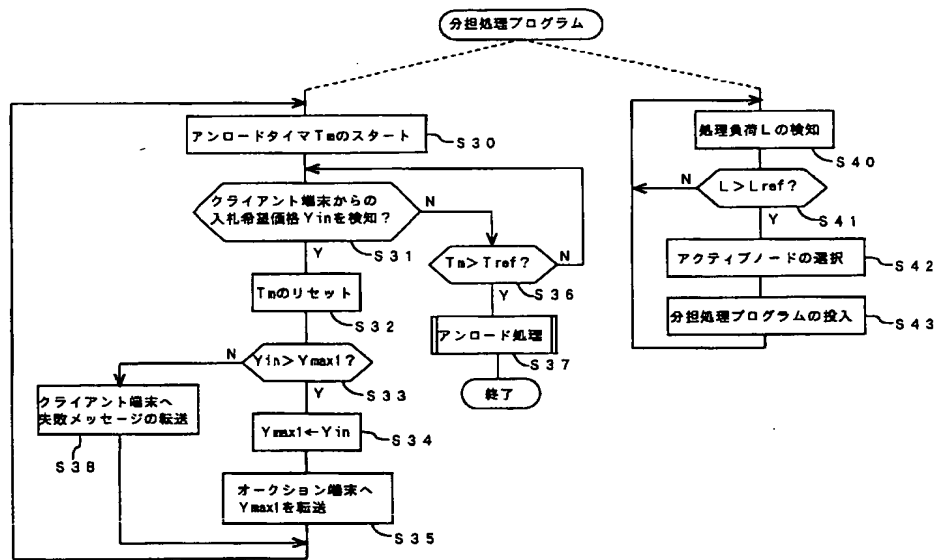


【図10】





【図7】



フロントページの続き

(51)Int.Cl. <sup>7</sup>	識別記号	F I	テーマコード* (参考)
G 0 6 F 15/16	6 2 0	G 0 6 F 17/60	1 4 8
17/60	1 4 8	9/06	4 2 0 J

(72)発明者 杉山 啓三  
 埼玉県上福岡市大原 2-1-15 株式会社  
 ケイディディ研究所内

(72)発明者 中尾 康二  
 埼玉県上福岡市大原 2-1-15 株式会社  
 ケイディディ研究所内

F ターム(参考) 5B045 GG04 HH02  
 5B049 AA02 AA04 BB36 EE04  
 5B076 BB06  
 5B089 GA11 GB02 JA11 JA40 JB07  
 JB22 KA06 KB04 KC28 KC60  
 MA03  
 5B098 AA10 GC10 GC16